On the Iterated Hairpin Completion

Steffen Kopecki

kopecki@fmi.uni-stuttgart.de

University of Stuttgart, Institute for Formal Methods in Computer Science (FMI),

Universitätsstraße 38, D-70569 Stuttgart

March 10, 2011

Abstract

The *(bounded) hairpin completion* and its iterated versions are operations on formal languages which have been inspired by the hairpin formation in DNA-biochemistry. The paper answers two questions asked in the literature about the iterated hairpin completion.

The first question is whether the class of regular languages is closed under iterated bounded hairpin completion. Here we show that this is true by providing a more general result which applies to all the classes of languages which are closed under finite union, intersection with regular sets, and concatenation with regular sets. In particular, all Chomsky classes and all standard complexity classes are closed under iterated bounded hairpin completion.

In the second part of the paper we address the question whether the iterated hairpin completion of a singleton is always regular. In contrast to the first question, this one has a negative answer. We exhibit an example of a singleton language whose iterated hairpin completion is not regular, actually it is not context-free, but context-sensitive.

Keywords: Formal languages, Finite automata, Hairpin completion, Bounded hairpin completion

1 Introduction

The hairpin completion is an operation on formal languages which is inspired by DNA-computing and biochemistry where it appears naturally in chemical reactions. It turned out that the corresponding operation on formal languages gives rise to very interesting and quite subtle decidability and computational problems. The focus of this paper is on these formal language theoretical results. However, let us sketch the biochemical origin of this operation first.

A *DNA strand* is a polymer composed of nucleotides which differ from each other by their bases A (adenine), C (cytosine), G (guanine), and T (thymine). For our purposes a strand can be seen as a finite sequence of bases. By *Watson-Crick base pairing* two base sequences can bind to each other if they are pairwise complementary, where A is complementary to T and C to G. The hairpin completion is best explained by Figure 1. By a sequence $\overline{w}$ we always mean to read w from right to left and to complement base by base, i.e., $\overline{a_1 \cdots a_n} = \overline{a_n} \cdots \overline{a_1}$. During a chemical process, called annealing, a strand which contains a sequence α and ends on the complementary sequence $\overline{\alpha}$, Fig. 1(a), can form an intramolecular base-pairing which is known as *hairpin* (in case α is not too short, say $|\alpha| \geq 10$), see Fig. 1(b). By complementing the unbound sequence γ, the *hairpin completion* arises, Fig. 1(c).

Hairpin completions of strands develop naturally during a technique called *Polymerase Chain Reaction* (PCR). The PCR is often used in DNA algorithms to amplify DNA strands with certain
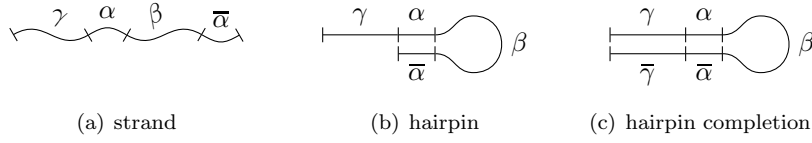
(a) strand (b) hairpin (c) hairpin completion

Figure 1: Hairpin completion of a DNA-strand.

properties. In many algorithms which use PCR the hairpin completions are by-products which cannot be used for the subsequent computation. Therefore, sets of strands which are unlikely to build hairpins (or lead to other *bad hybridizations*) have been examined in many papers, see e.g., [2, 4, 5, 10, 11].

On the other hand, some DNA-based computations rely on the fact that DNA strands can form hairpins. An example are algorithms using the *Whiplash PCR* in which strands are designed to build hairpins. This technique can be used to solve combinatorial problems, including NP-complete ones like SATISFIABILITY and HAMILTONIAN PATH, see [6, 18, 19].

On an abstract level a strand can be seen as a word and a (possibly infinite) set of strands is a language. The hairpin completion of formal languages has been introduced in [1] by Cheptea, Martín-Vide, and Mitrana. In several papers the hairpin completion and some familiar operations have been studied, see e.g., [1, 3, 14–17]. The focus of this paper is on closure properties of language classes concerning the iterated versions of the hairpin completion and the bounded hairpin completion. For the latter operation we assume the length of the γ-part to be bounded. This variant of the hairpin completion was introduced and analyzed in [8, 9] by Ito, Leupold, Manea, and Mitrana. A formal definition of both operations is given in Section 2.1.

In [1] the closure properties of different language classes under the non-iterated and iterated hairpin completion have been analyzed. It follows that neither regular nor context-free languages are closed under hairpin completion whereas the family of context-sensitive languages is closed under this function. Actually, from [1] we can derive that the class DSPACE(f) (resp. the class NSPACE(f)) is closed under hairpin completion (resp. closed under iterated hairpin completion) for every function $f \in \Omega(\log)$. (By the class DSPACE(f) (resp. NSPACE(f)) we mean, as usual, the class of languages that can be accepted by a deterministic (resp. non-deterministic) Turing machine which uses $f(n)$ work space on input length n.) In particular, the class of context-sensitive languages is closed under iterated hairpin completion, too. Furthermore, if we apply the iterated hairpin completion to a regular (resp. context-free) language we stay inside NL($=$ NSPACE(log)) (resp. NSPACE($\log^2$), by Lewis, Stearns, and Hartmanis [13]) which is in terms of space complexity far below the class of deterministic context-sensitive languages.

The situation changes if we consider the bounded hairpin completion, which can be seen as a weaker variant of the hairpin completion. All classes in the Chomsky Hierarchy are closed under bounded hairpin completion and the classes of context-free, context-sensitive, and recursively enumerable languages are closed under the iterated operation, see [8, 9]. But the status for regular languages remained unknown and was stated as an open problem in [9]. In Section 3 we solve this problem. We state a general representation for the iterated bounded hairpin completion of a formal language using the operations union, intersection with regular sets, and concatenation with regular sets (Theorem 3.1). As a consequence all language classes which are closed under these basic operations are also closed under iterated bounded hairpin completion.

Furthermore, for a given non-deterministic finite automaton (NFA) accepting a language L, we give exponential lower and upper bounds for the size of an NFA accepting the iterated bounded hairpin completion of L in Theorem 4.1. Thus, if we ignore constants, the NFA leads us to a linear

time membership test for the iterated bounded hairpin completion of a fixed regular language. This improves a quadratic bound which was known before. Indeed, the best known time complexity of the membership problem for the iterated (unbounded) hairpin completion of a regular language L is still quadric time by an algorithm from [14]. See Section 4 for a more detailed discussion.

The class of iterated hairpin completions of singletons (HCS) has been investigated in [17] by Manea, Mitrana, and Yokomori (which is the journal version of a paper that appeared at AFL 2008). Obviously, HCS is included in the class of context-sensitive languages. However, the questions if HCS contains non-regular or non-context-free languages has been unsolved. In Section 5 we answer this question by stating a singleton whose iterated hairpin completion is not context-free.

This paper is the journal version of results which appeared as a poster at DLT 2010, [12].

2 Definitions and Notation

We assume the reader to be familiar with the fundamental concepts of formal language and automata theory, see [7].

An *alphabet* is a finite set of *letters*. In this paper the alphabet is always Σ. The set of words over Σ is denoted by Σ^*, as usual, and the *empty word* is denoted by ε. We consider Σ with an *involution*; this is a bijection $\overline{} : \Sigma \to \Sigma$ such that $\overline{\overline{a}} = a$ for all letters $a \in \Sigma$ (in DNA-biochemistry: $\Sigma = \{A, C, G, T\}$ with $\overline{A} = T$ and $\overline{C} = G$). We extend the involution to words $w = a_1 \cdots a_n$ by $\overline{w} = \overline{a_n} \cdots \overline{a_1}$. (Just like taking inverses in groups.) For a formal language L by $\overline{L}$ we denote the language $\{\overline{w} \mid w \in L\}$.

Given a word w, we denote by $|w|$ its length. For a length bound $\ell \geq 0$ the set $\Sigma^{\leq \ell}$ contains all words of length at most ℓ. If $w = xyz$ for some $x, y, z \in \Sigma^*$, then x, y, and z are called *prefix*, *factor*, and *suffix* of the word w, respectively. For the prefix relation we also use the notation $x \leq w$. Note that if z is a suffix of w, then $\overline{z}$ is a prefix of $\overline{w}$ (or $\overline{z} \leq \overline{w}$).

A common way to describe regular languages are *non-deterministic finite automata* (NFAs). An NFA A is a tuple (Q, Σ, E, I, F) where Q is the *finite set of states*, $I \subseteq Q$ is the set of *initial states*, $F \subseteq Q$ is the set of *final states*, and $E \subseteq Q \times \Sigma \times Q$ is the set of *labelled edges* or *transitions*. The language accepted by the automaton, denoted by $L(A)$, contains all words w such that there is a path labelled by w which leads from an initial state to a final state. By the size of an NFA we mean the number of states $|Q|$.

2.1 The Hairpin Completion

Let $w \in \Sigma^*$ be a word. If w has a factorization $w = \gamma \alpha \beta \overline{\alpha}$, it can form a hairpin and $\gamma \alpha \beta \overline{\alpha} \overline{\gamma}$ is a *right hairpin completion* of w (again, see Figure 1). Since a hairpin in biochemistry is stable only if α is long enough, we fix a constant $k \geq 1$ and ask $|\alpha| = k$. (Note that the definition does not change if we ask $|\alpha| \geq k$.)

Symmetrically, if w has a factorization $\alpha \beta \overline{\alpha} \overline{\gamma}$ with $|\alpha| = k$, then $\gamma \alpha \beta \overline{\alpha} \overline{\gamma}$ is a *left hairpin completion* of w. For the bounded hairpin completion we assume that the length of the factor γ is bounded by some constant.

The hairpin completion of a formal language L is the union of all hairpin completions of all words in L. Before we state the formal definition of the unbounded and bounded hairpin completion of a language, we introduce a more general variant of the hairpin completion, namely the *parameterized hairpin completion*. The parameterized hairpin completion covers the other operations as special cases.

Let $\ell, r \in \mathbb{N} \cup \{\infty\}$ be two length bounds and let L be a formal language. Considering a left hairpin completion with the factorization $\gamma \alpha \beta \overline{\alpha} \overline{\gamma}$ as above, then the bound ℓ limits the length of γ; respectively, the bound r limits the length of γ in a right hairpin completion. For a word $\alpha \in \Sigma^k$ the parameterized hairpin completion is defined as

$$\mathcal{H}_\alpha(L, \ell, 0) = \bigcup_{\gamma \in \Sigma^{\leq \ell}} \gamma \left(\alpha \Sigma^* \overline{\alpha} \overline{\gamma} \cap L \right)$$

$$\mathcal{H}_\alpha(L, 0, r) = \bigcup_{\gamma \in \Sigma^{\leq r}} \left(\gamma \alpha \Sigma^* \overline{\alpha} \cap L \right) \overline{\gamma}$$

$$\mathcal{H}_\alpha(L, \ell, r) = \mathcal{H}_\alpha(L, \ell, 0) \cup \mathcal{H}_\alpha(L, 0, r).$$

For the constant k we define

$$\mathcal{H}_k(L, \ell, r) = \bigcup_{\alpha \in \Sigma^k} \mathcal{H}_\alpha(L, \ell, r).$$

In the unbounded case we distinguish two operations: The *(two-sided) hairpin completion* is defined as $\mathcal{H}_k(L) = \mathcal{H}_k(L, \infty, \infty)$ and the *right-sided hairpin completion* is defined as $\mathcal{RH}_k(L) = \mathcal{H}_k(L, 0, \infty)$. For the latter case we allow right hairpin completions, only. In the same way we might define the left-sided hairpin completion of a language, but for convenience we will treat the right-sided operation, only, and also refer to it as the *one-sided hairpin completion*. It is plain, that our results also hold for the left-sided case.

The *bounded hairpin completion* $\mathcal{H}(L, m, m)$ arises if we choose the same finite bound $m \in \mathbb{N}$ for left and right hairpin completions.

Note that if both bounds ℓ, r are finite and L is regular, then the parameterized hairpin completion $\mathcal{H}_k(L, \ell, r)$ is regular as well. This does not hold if $\ell = \infty$ or $r = \infty$ as one of the unions becomes infinite. It is known that the unbounded hairpin completion of a regular language is not necessarily regular but always linear context-free, see e.g., [1].

In this paper we examine the iterated versions of the operations we defined so far. The iterated hairpin completion of a language L contains all words which belong to a sequence $w_0, \ldots, w_n$ where $w_0 \in L$ and where w_i is a right or left hairpin completion of w_{i-1} and the bound r (resp. ℓ) applies for all i such that $1 \leq i \leq n$. More formal, let $\ell, r \in \mathbb{N} \cup \{\infty\}$ and

$$\mathcal{H}_\alpha^0(L, \ell, r) = L, \qquad \mathcal{H}_\alpha^i(L, \ell, r) = \mathcal{H}_\alpha(\mathcal{H}_\alpha^{i-1}(L, \ell, r), \ell, r),$$
$$\mathcal{H}_k^0(L, \ell, r) = L, \qquad \mathcal{H}_k^i(L, \ell, r) = \mathcal{H}_k(\mathcal{H}_k^{i-1}(L, \ell, r), \ell, r)$$

for $i \geq 1$. The iterated parameterized hairpin completion of L is the union

$$\mathcal{H}_\alpha^*(L, \ell, r) = \bigcup_{i \geq 0} \mathcal{H}_\alpha^i(L, \ell, r) \qquad \text{resp.} \qquad \mathcal{H}_k^*(L, \ell, r) = \bigcup_{i \geq 0} \mathcal{H}_k^i(L, \ell, r).$$

If a word z is included in $\mathcal{H}_k^i(\{w\}, \ell, r)$, we say z is an *i-iterated hairpin completion* of w, and if $z \in \mathcal{H}_k^*(\{w\}, \ell, r)$, we say z is an *iterated hairpin completion* of w. (It will be clear from the context which length bounds apply.)

The iterated unbounded hairpin completions are denoted by $\mathcal{H}_k^*(L) = \mathcal{H}_k^*(L, \infty, \infty)$ and $\mathcal{RH}_k^*(L) = \mathcal{H}_k^*(L, 0, \infty)$.

Example. Figure 2 shows a 3-iterated hairpin completion of $\alpha u \overline{\alpha} v \alpha$ where $|\alpha| = k$. In each step the dotted part is the newly created prefix or suffix.
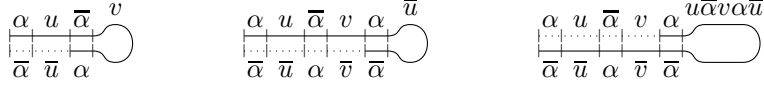
Figure 2: Example for the iterated hairpin completion.

3 The Iterated Bounded Hairpin Completion

In this section we will give a general representation for the iterated parameterized hairpin completion with finite bounds. Our main result is the proof of the following theorem which can be found in Section 3.2.

Theorem 3.1. *Let L be a formal language and $\ell, r \in \mathbb{N}$. The iterated parameterized hairpin completion $\mathcal{H}_k^*(L, \ell, r)$ can be effectively represented by an expression using L and the operations union, intersection with regular sets, and concatenation with regular sets.*

Consequentially, all language classes which are closed under these operations are also closed under iterated parameterized hairpin completion with finite bounds, and if the closure under all three operations is effective, then the closure under iterated parameterized hairpin completion with finite bounds is effective, too; this applies to all four Chomsky classes. From [9] it is known that the classes of context-free, context-sensitive, and recursively enumerable languages are closed under iterated bounded hairpin completion, but the status for regular languages was unknown. Since the iterated bounded hairpin completion is a special case of the iterated parameterized hairpin completion with finite bounds we can answer this question now.

Corollary 3.2. *Let $\mathcal{C}$ be a class of languages. If $\mathcal{C}$ is closed under union, intersection with regular sets, and concatenation with regular sets, then $\mathcal{C}$ is also closed under iterated bounded hairpin completion. Moreover, if $\mathcal{C}$ is effectively closed under union, intersection with regular sets, and concatenation with regular sets, then the closure under iterated bounded hairpin completion is effective.*

In particular, the class of regular languages is effectively closed under iterated bounded hairpin completion.

The next two sections are devoted to the proof of Theorem 3.1. First we introduce the important concept of α-prefixes.

3.1 α-Prefixes

Let α be a word of length k. For $v, w \in \Sigma^*$ we say v is an α-prefix of w if $v\alpha \leq w$. We denote the set of all α-prefixes of length at most ℓ by

$$\mathcal{P}_\alpha(w, \ell) = \{v \mid v\alpha \leq w \wedge |v| \leq \ell\}.$$

The idea behind this notation is: For a word $w \in \alpha\Sigma^*\overline{\alpha}$ with $|w| - k \geq \ell, r$, the set of (non-iterated) parameterized hairpin completions of w is given by

$$\mathcal{H}_\alpha(\{w\}, \ell, 0) = \mathcal{P}_\alpha(\overline{w}, \ell)w \qquad \text{and} \qquad \mathcal{H}_\alpha(\{w\}, 0, r) = w\overline{\mathcal{P}_\alpha(w, r)}.$$

In the following proof we are interested in α-prefixes of words which have α as a prefix. This leads to some useful properties.

Lemma 3.3. *Let $\alpha \in \Sigma^k$, $\ell \in \mathbb{N}$, and $w \in \alpha\Sigma^*$.*

1. For all $v \in \mathcal{P}_\alpha(w, \ell)$ we have $\alpha \le v\alpha$.

2. For all $u, v \in \mathcal{P}_\alpha(w, \ell)$ we have

$$|u| \le |v| \iff u\alpha \le v\alpha \iff u \in \mathcal{P}_\alpha(v\alpha, \ell).$$

3. If $v\alpha$ is a prefix of some word in $\mathcal{P}_\alpha(w, \ell)^*\alpha$, then $v \in \mathcal{P}_\alpha(w, \ell)^*$.

Proof. If two words x, y are prefixes of w and $|x| \le |y|$, then $x \le y$. This yields properties 1 and 2.

For property 3 let $v\alpha \le x_1 \cdots x_m\alpha$ where $x_1, \ldots, x_m \in \mathcal{P}_\alpha(w, \ell)$. We can factorize $v = x_1 \cdots x_{i-1}y$ such that $y \le x_i$ for some i with $1 \le i \le m$. By property 1 and induction, we see that α is a prefix of $x_{i+1} \cdots x_m\alpha$ and hence $y\alpha \le x_i\alpha \le w$ which implies $y \in \mathcal{P}_\alpha(w, \ell)$ and, moreover, $v \in \mathcal{P}_\alpha(w, \ell)^*$. $\square$

3.2 Proof of Theorem 3.1

Let L be a formal language and $\ell, r \in \mathbb{N}$. We will state a representation for $\mathcal{H}_k^*(L, \ell, r)$ using L and the operations union, intersection with regular sets, and concatenation with regular sets.

Let us begin with a basic observation. Every word w which is a hairpin completion of some other word has a factorization $w = \delta\beta\bar{\delta}$ with $|\delta| \ge k$, therefore, the prefix of w of length k and the suffix of w of length k are complementary. Let us call this prefix α, hence, we have $w \in \alpha\Sigma^*\bar{\alpha}$. Every word which is a right hairpin completion of w has still the prefix α and since the suffix of length k is complementary, it has the suffix $\bar{\alpha}$ as well. For left hairpin completions we have a symmetric argument and, by induction, every word which is an iterated hairpin completion of w has prefix α and suffix $\bar{\alpha}$.

Thus, we can split up the (non-iterated) parameterized hairpin completion $\mathcal{H}_k(L, \ell, r)$ into finitely many languages $L_\alpha = \mathcal{H}_k(L, \ell, r) \cap \alpha\Sigma^*\bar{\alpha}$ where $\alpha \in \Sigma^k$, and each of them has a effective representation using L and the operations union, intersection with regular sets, and concatenation with regular sets. Moreover,

$$\mathcal{H}_k^*(L_\alpha, \ell, r) = \mathcal{H}_\alpha^*(L_\alpha, \ell, r) \subseteq \alpha\Sigma^*\bar{\alpha}$$

and the iterated parameterized hairpin completion equals

$$\begin{aligned} \mathcal{H}_k^*(L, \ell, r) &= L \cup \mathcal{H}_k^*(\mathcal{H}_k(L, \ell, r), \ell, r) \\ &= L \cup \mathcal{H}_k^*\Big(\bigcup_{\alpha \in \Sigma^k} L_\alpha, \ell, r \Big) \\ &= L \cup \bigcup_{\alpha \in \Sigma^k} \mathcal{H}_\alpha^*(L_\alpha, \ell, r). \end{aligned}$$

Henceforth, let $\alpha \in \Sigma^k$ be fixed. In order to prove Theorem 3.1 we will state a suitable representation for $\mathcal{H}_\alpha^*(L_\alpha, \ell, r)$. For the rest of the proof we will heavily rely on the fact that every word in $\mathcal{H}_\alpha^*(L_\alpha, \ell, r)$ has the prefix α and the suffix $\bar{\alpha}$. The representation is defined recursively. We have

$$\mathcal{H}_\alpha^*(L_\alpha, 0, 0) = L_\alpha.$$

By symmetry, we may assume that $\ell \ge r$ and $\ell \ge 1$. We will state a representation for $\mathcal{H}_\alpha^*(L_\alpha, \ell, r)$ using $\mathcal{H}_\alpha^*(L_\alpha, \ell - 1, r)$ and the operations union, intersection with regular sets, and concatenation with regular sets. Therefore, consider a word

$$z \in \mathcal{H}_\alpha^*(L_\alpha, \ell, r) \setminus \mathcal{H}_\alpha^*(L_\alpha, \ell - 1, r).$$

For some $n \geq 1$ there is a sequence $w_0, \dots, w_n = z$ where $w_0 \in L_\alpha$ and for all i such that $1 \leq i \leq n$ either w_i is a left hairpin completion of w_{i-1} and $|w_i| \leq |w_{i-1}| + \ell$ or w_i is a right hairpin completion of w_{i-1} and $|w_i| \leq |w_{i-1}| + r$. Furthermore, there is an index $j \geq 1$ such that $w_{j-1} = w \in \mathcal{H}_\alpha^*(L_\alpha, \ell - 1, r)$ and $w_j = vw \notin \mathcal{H}_\alpha^*(L_\alpha, \ell - 1, r)$. Note that this implies $|v| = \ell$ and $w \in \alpha \Sigma^* \overline{\alpha v}$. Let $s = n - j$ and consider the factorization

$$ z = x_s \cdots x_1 vw \overline{y_1} \cdots \overline{y_s} $$

where $x_i \cdots x_1 vw \overline{y_1} \cdots \overline{y_i} = w_{j+i}$ and either

1. $y_i = \varepsilon$, $|x_i| \leq \ell$, and $x_i \alpha \leq y_{i-1} \cdots y_1 v\alpha$ or

2. $x_i = \varepsilon$, $|y_i| \leq r$, and $y_i \alpha \leq x_{i-1} \cdots x_1 v\alpha$.

for all i such that $0 \leq i \leq s$.

The crucial point is that vw has the prefix $v\alpha$, the suffix $\overline{\alpha v}$, and $|v| = \ell \geq r$. Therefore, the factors $x_1, \dots, x_s$ and $y_1, \dots, y_s$ are controlled by the triple (v, ℓ, r) in the following way.

Lemma 3.4. $x_i \in \mathcal{P}_\alpha(v\alpha, \ell)^*$ and $y_i \in \mathcal{P}_\alpha(v\alpha, r)^*$ for all i such that $1 \leq i \leq s$.

Proof. We prove the claim by induction on i. Let i such that $1 \leq i \leq s$. Our induction hypothesis is $x_j \in \mathcal{P}_\alpha(v\alpha, \ell)^*$ and $y_j \in \mathcal{P}_\alpha(v\alpha, r)^*$ for all j such that $1 \leq j < i$. We distinguish between the two cases above:

1. We have $y_i = \varepsilon \in \mathcal{P}_\alpha(v\alpha, r)^*$ and, by induction hypothesis,

$$ x_i \alpha \leq y_{i-1} \cdots y_1 v\alpha \in \mathcal{P}_\alpha(v\alpha, r)^* v\alpha \subseteq \mathcal{P}_\alpha(v\alpha, \ell)^* \alpha. $$

   In combination with Lemma 3.3 this yields $x_i \in \mathcal{P}_\alpha(v\alpha, \ell)^*$.

2. We have $x_i = \varepsilon \in \mathcal{P}_\alpha(v\alpha, \ell)^*$ and

$$ y_i \alpha \leq x_{i-1} \cdots x_1 v\alpha \in \mathcal{P}_\alpha(v\alpha, \ell)^* \alpha, $$

   hence $y_i \in \mathcal{P}_\alpha(v\alpha, \ell)^*$. Since $|y_i| \leq r$, all factors of y_i are at most of length r, too, and $y_i \in \mathcal{P}_\alpha(v\alpha, r)^*$. $\qquad\square$

For $u \in \Sigma^\ell$ let us define the language

$$ \mathcal{L}_\alpha(u, \ell, r) = \mathcal{P}_\alpha(u\alpha, \ell)^* u \left(\mathcal{H}_\alpha^*(L_\alpha, \ell - 1, r) \cap \alpha \Sigma^* \overline{\alpha u} \right) \overline{\mathcal{P}_\alpha(u\alpha, r)}^*. $$

Note that, by induction, for every u the representation for $\mathcal{L}_\alpha(u, \ell, r)$ is effectively given. By Lemma 3.4, the word z is included in $\mathcal{L}_\alpha(v, \ell, r)$ and for every word $z' \in \mathcal{H}_\alpha^*(L_\alpha, \ell, r) \setminus \mathcal{H}_\alpha^*(L_\alpha, \ell - 1, r)$ it exists $v' \in \Sigma^\ell$ such that $z' \in \mathcal{L}_\alpha(v', \ell, r)$. Therefore,

$$ \mathcal{H}_\alpha^*(L_\alpha, \ell, r) \subseteq \mathcal{H}_\alpha^*(L_\alpha, \ell - 1, r) \cup \bigcup_{u \in \Sigma^\ell} \mathcal{L}_\alpha(u, \ell, r) $$

and for the right hand side we have an effective representation. Of course, we intend to replace the inclusion by an equality sign.

Lemma 3.5. $\mathcal{L}_\alpha(u, \ell, r) \subseteq \mathcal{H}_\alpha^*(L_\alpha, \ell, r)$ for all $u \in \Sigma^\ell$.

Proof. We start by proving a special case of the claim that is successfully used later to derive the result. Consider a word w' together with the factorization

$$w' = x_m \cdots x_1 w \overline{y_1} \cdots \overline{y_n}$$

with $m \geq 0$, $n \geq 1$ and where for some word $u \in \Sigma^*$

1. $w \in \mathcal{H}_\alpha^*(L_\alpha, \ell, r) \cap u\alpha\Sigma^*\overline{\alpha}\overline{u}$,

2. $x_1, \ldots, x_m \in \mathcal{P}_\alpha(u\alpha, \ell)$,

3. $y_1, \ldots, y_n \in \mathcal{P}_\alpha(u\alpha, r)$, and

4. $m = 0$ or $|y_j| \leq |x_m|$ for all j such that $1 \leq j \leq n$.

We claim $w' \in \mathcal{H}_\alpha^*(L_\alpha, \ell, r)$, too. Indeed, if $m = 0$, it is plain that w' is an n-iterated right hairpin completion of w. Otherwise $x_m \cdots x_1 w$ is an m-iterated left hairpin completion of w. By the fourth property and Lemma 3.3, we have $y_1, \ldots, y_n \in \mathcal{P}_\alpha(x_m\alpha, r)$. Hence, w' is an n-iterated right hairpin completion of $x_m \cdots x_1 w$ and we conclude $w' \in \mathcal{H}_\alpha^*(L_\alpha, \ell, r)$.

Now, let $u \in \Sigma^\ell$ and $z \in \mathcal{L}_\alpha(u, \ell, r)$. There is a factorization

$$z = x_s \cdots x_1 w \overline{y_1} \cdots \overline{y_t}$$

where

1. $w \in u\left(\mathcal{H}_\alpha^*(L, \ell-1, r) \cap \alpha\Sigma^*\overline{\alpha}\overline{u}\right) \subseteq \mathcal{H}_\alpha^*(L_\alpha, \ell, r) \cap u\alpha\Sigma^*\overline{\alpha}\overline{u}$,

2. $x_1, \ldots, x_s \in \mathcal{P}_\alpha(u\alpha, \ell)$, and

3. $y_1, \ldots, y_t \in \mathcal{P}_\alpha(u\alpha, r)$.

If $t = 0$, the word z is an s-iterated left hairpin completion of w. Otherwise, let $n \geq 1$ be the maximal index such that $|y_n| \geq |y_j|$ for all $1 \leq j \leq t$, and let m be the maximal index such that $|y_n| \leq |x_m|$ or 0 if no such index exists. Let $w' = x_m \cdots x_1 w \overline{y_1} \cdots \overline{y_n}$. Note that w' satisfies the conditions of the special case we discussed above and hence $w' \in \mathcal{H}_\alpha^*(L_\alpha, \ell, r)$.

With $u' = y_n$ we obtain

$$z = x_s \cdots x_{m+1} w' \overline{y_{n+1}} \cdots \overline{y_t}$$

where, by the choice of n, m and by Lemma 3.3,

1. $w' \in \mathcal{H}_\alpha^*(L_\alpha, \ell, r) \cap u'\alpha\Sigma^*\overline{\alpha}\overline{u'}$,

2. $x_{m+1}, \ldots, x_s \in \mathcal{P}_\alpha(u'\alpha, \ell)$, and

3. $y_{n+1}, \ldots, y_t \in \mathcal{P}_\alpha(u'\alpha, r)$.

At this point we may continue inductively and deduce $z \in \mathcal{H}_\alpha^*(L_\alpha, \ell, r)$. $\square$

The previous lemma tells us, if $\ell \geq r$, the iterated parameterized hairpin completion of L_α can be represented by

$$\mathcal{H}_\alpha^*(L_\alpha, \ell, r) = \mathcal{H}_\alpha^*(L_\alpha, \ell-1, r) \cup \bigcup_{u \in \Sigma^\ell} \mathcal{L}_\alpha(u, \ell, r).$$

Symmetrically, if $r > \ell$, let us define

$$\mathcal{R}_\alpha(u, \ell, r) = \mathcal{P}_\alpha(u\alpha, \ell)^* \left(\mathcal{H}_\alpha^*(L_\alpha, \ell, r-1) \cap u\alpha\Sigma^*\overline{\alpha}\right) \overline{u}\,\overline{\mathcal{P}_\alpha(u\alpha, r)}^*.$$

The iterated parameterized hairpin completion of L_α can be represented by

$$\mathcal{H}_\alpha^*(L_\alpha, \ell, r) = \mathcal{H}_\alpha^*(L_\alpha, \ell, r-1) \cup \bigcup_{u \in \Sigma^r} \mathcal{R}_\alpha(u, \ell, r).$$

We conclude, the iterated parameterized hairpin completion of a language L can be represented by an expression using L and the operations union, intersection with regular sets, and concatenation with regular sets.

4 The size of NFAs accepting iterated parameterized hairpin completions

Let L be a regular language and $\ell, r \in \mathbb{N}$ be finite bounds. In this section we analyze the size of NFAs accepting the iterated parameterized hairpin completion $\mathcal{H}_k^*(L, \ell, r)$ with respect to the size of an NFA accepting L and the bounds ℓ and r. By the size of an NFA we mean its number of states. Recall that k is treated as a constant. (Assuming $k \le \ell$ or $k \le r$ would induce the same complexity, but this is not shown here.) Our results are the following.

Theorem 4.1.

1. *Let $m \ge 1$. There is a regular language L such that neither the language $\mathcal{H}_k(L, m, m)$ nor the language $\mathcal{H}_k^*(L, m, m)$ can be detected by an NFA with less than 2^m states.*

2. *Let L be a regular language which is accepted by an NFA of size n. Let $\ell, r \in N$ and let $m = \max\{\ell, r\}$. There is an NFA accepting the iterated parameterized hairpin completion $\mathcal{H}_k^*(L, \ell, r)$ whose size is in $2^{\mathcal{O}(m^2)} n$.*

Proof of 1. Let $\Sigma = \{a, \overline{a}, b, \overline{b}, c, \overline{c}\}$ and $L = c\{\overline{a}, \overline{b}\}^* a^k \overline{a}^k$. For any word $w \in L$ there is no possibility of building a left hairpin and the only possible right hairpin is to bind the suffix $\overline{a}^k$ to a^k if $|w| \le m + 2k$. Therefore, we have

$$\mathcal{H}_k(L, m, m) = \bigcup_{v \in \{\overline{a}, \overline{b}\}^{\le m-1}} cva^k \overline{a}^k \overline{v} \overline{c}.$$

Now let $w = cva^k \overline{a}^k \overline{v} \overline{c}$ with $v \in \{\overline{a}, \overline{b}\}^{\le m-1}$. The only way to build a hairpin is to bind its prefix to its suffix, hence

$$\mathcal{H}_k^*(L, m, m) = L \cup \mathcal{H}_k(L, m, m).$$

We claim that an NFA accepting $\mathcal{H}_k(L, m, m)$ or $\mathcal{H}_k^*(L, m, m)$ has a size of at least 2^m. We prove the claim for the language $\mathcal{H}_k(L, m, m)$; the argumentation for $\mathcal{H}_k^*(L, m, m)$ is exactly the same.

Consider an NFA accepting $\mathcal{H}_k(L, m, m)$ and let Q denote its set of states. For a word $u \in \Sigma^*$ we denote by $P(u) \subseteq Q$ the set of states which are reachable from an initial state with a path labelled by u. Now let $v \in \{\overline{a}, \overline{b}\}^{\le m-1}$. Since $cva^k \overline{a}^k \overline{v} \overline{c} \in \mathcal{H}_k(L, m, m)$, there is a state $q \in P(cva^k \overline{a}^k)$ such that a path from q to a final state exists which is labelled by $\overline{v} \overline{c}$. For all words $u \in \{\overline{a}, \overline{b}\}^{\le m-1}$ with $u \ne v$ the state q does not belong to $P(cua^k \overline{a}^k)$ because $cua^k \overline{a}^k \overline{v} \overline{c} \notin \mathcal{H}_k(L, m, m)$. Each word $v \in \{\overline{a}, \overline{b}\}^{\le m-1}$ yields such a state q, they are mutually different, and none of them is an initial state (as $\overline{v} \overline{c} \notin \mathcal{H}_k(L, m, m)$). Therefore, the number of states $|Q|$ has to be greater than $\left|\{\overline{a}, \overline{b}\}^{\le m-1}\right| = 2^m - 1$. $\square$

In order to prove the second claim of Theorem 4.1 we implicitly use some well-known constructions of NFAs which accept concatenation, union, or intersection of regular languages. Consider two NFAs which accept the languages L_1, L_2 and which are of size n_1, n_2, respectively. There is an NFA accepting the concatenation $L_1 L_2$ which is of size $n_1 + n_2$, an NFA accepting the union $L_1 \cup L_2$ which is of size $n_1 + n_2$, and an NFA accepting the intersection $L_1 \cap L_2$ which is of size $n_1 \cdot n_2$. For details on how these NFAs are constructed see, e.g., [7].

Proof of 2. Let L be a regular language which is accepted by an automaton of size n and let $\ell, r \in \mathbb{N}$. The parameterized hairpin completion of L is given by

$$\mathcal{H}_k(L, \ell, r) = \bigcup_{\alpha \in \Sigma^k} \bigcup_{\gamma \in \Sigma^{\leq \ell}} \gamma(\alpha \Sigma^* \overline{\alpha} \overline{\gamma} \cap L) \cup \bigcup_{\alpha \in \Sigma^k} \bigcup_{\gamma \in \Sigma^{\leq r}} (\gamma \alpha \Sigma^* \overline{\alpha} \cap L) \overline{\gamma}.$$

For $\gamma, \alpha \in \Sigma^*$ there is an NFA accepting $\gamma(\alpha \Sigma^* \overline{\alpha} \overline{\gamma} \cap L)$ which has a size in $\mathcal{O}(|\gamma \alpha| \cdot n)$. Hence, the parameterized hairpin completion of L can be accepted by an NFA which has a size in $\mathcal{O}(|\Sigma|^m m \cdot n) \subseteq 2^{\mathcal{O}(m)} n$ where $m = \max\{\ell, r\}$.

For $\alpha \in \Sigma^k$ the language $L_\alpha = \mathcal{H}_k(L, \ell, r) \cap \alpha \Sigma^* \overline{\alpha}$ can also be accepted by an NFA which has a size in $2^{\mathcal{O}(m)} n$. Let $N_{i,j}$ denote the minimal size of an NFA accepting $\mathcal{H}_\alpha^*(L_\alpha, i, j)$ for $i, j \in \mathbb{N}$. Since $\mathcal{H}_k(L_\alpha, 0, 0) = L_\alpha$, we have $N_{0,0} \in 2^{\mathcal{O}(m)} n$. For $i \geq j$ let us recall that

$$\mathcal{H}_\alpha^*(L_\alpha, i, j) = \mathcal{H}_\alpha^*(L_\alpha, i-1, j) \cup \bigcup_{u \in \Sigma^i} \mathcal{L}_\alpha(u, i, j),$$

$$\mathcal{L}_\alpha(u, i, j) = \mathcal{P}_\alpha(u\alpha, \ell)^* u \left(\mathcal{H}_\alpha^*(L_\alpha, i-1, j) \cap \alpha \Sigma^* \overline{\alpha u}\right) \overline{\mathcal{P}_\alpha(u\alpha, r)}^*.$$

The size of a minimal NFA accepting $\mathcal{L}_\alpha(u, i, j)$ is in $\mathcal{O}(i \cdot N_{i-1,j})$ whence

$$N_{i,j} \in \mathcal{O}(|\Sigma|^i i \cdot N_{i-1,j}) \subseteq 2^{\mathcal{O}(i)} N_{i-1,j}.$$

Symmetrically, for $j > i$ we have $N_{i,j} \in 2^{\mathcal{O}(i)} N_{i,j-1}$. By unfolding the recursion we obtain

$$N_{\ell,r} \in \prod_{i=1}^{\ell} 2^{\mathcal{O}(i)} \cdot \prod_{j=1}^{r} 2^{\mathcal{O}(j)} \cdot 2^{\mathcal{O}(m)} n = \prod_{i=1}^{m} 2^{\mathcal{O}(i)} \cdot n = 2^{\mathcal{O}(\sum_{i=1}^{m} i)} n = 2^{\mathcal{O}(m^2)} n.$$

Now, the iterated parameterized hairpin completion is given by

$$\mathcal{H}_k^*(L, \ell, r) = L \cup \bigcup_{\alpha \in \Sigma^k} \mathcal{H}_\alpha^*(L_\alpha, \ell, r).$$

and there is an NFA accepting $\mathcal{H}_k^*(L, \ell, r)$ which has a size in $\mathcal{O}(N_{\ell,r} + n) \subseteq 2^{\mathcal{O}(m^2)} n$. $\square$

Statement 2 of Theorem 4.1 also yields an algorithm to solve the membership problem for the iterated bounded hairpin completion of a regular language.

Corollary 4.2. *Let L be a regular language, given by an NFA of size n, and let $\ell, r \in \mathbb{N}$. The problem whether an input word w belongs to $\mathcal{H}_k^*(L, \ell, r)$ can be decided in linear time $c \cdot |w|$, where the constant c depends on the size n and the bounds ℓ, r. More precisely, for $m = \max\{\ell, r\}$ we have $c \in 2^{\mathcal{O}(m^2)} n^2$.*

Proof. Following the proof of Statement 2 of Theorem 4.1, we can construct an NFA $A = (Q, \Sigma, E, I, F)$ accepting the iterated hairpin completion $\mathcal{H}_k^*(L, \ell, r)$ which is of a size in $2^{\mathcal{O}(m^2)} n$.

Let us denote the size of this NFA by N. Note that the construction can be preformed in time $\mathcal{O}(|E|) \subseteq \mathcal{O}(N^2) \subseteq 2^{\mathcal{O}(m^2)} n^2$.

The input w can be accepted by an online power-set construction of the NFA A: We start with the set of states $P_0 = I$. When we read the i-th letter a of the input w we construct the set P_i by following all outgoing edges of states in P_{i-1} which are labelled by a. As every state in P_{i-1} has at most N outgoing edges labelled by a, one step can be performed in $\mathcal{O}(N^2) \subseteq 2^{\mathcal{O}(m^2)} n^2$ time. The algorithm stops after w is read and $P_{|w|}$ is computed. The input w belongs to $\mathcal{H}_k^*(L, \ell, r)$ if and only if $P_{|w|}$ contains a final state from F. $\qquad\square$

So far, the best known time complexity of the membership problem for the iterated (un-bounded) hairpin completion of a regular language L is quadratic with respect to the length of the input word, by an algorithm from [14]. This algorithm can easily be adapted to solve the membership problem for the iterated bounded hairpin completion in quadratic time. Hence, if we measure the time complexity with respect to the length of the input word only, we have an improvement from quadratic to linear time (in the bounded case).

5   The Iterated Hairpin Completion of Singletons

The class of iterated hairpin completions of singletons is defined as

$$\mathrm{HCS}_k = \{\mathcal{H}_k^*(\{w\}) \mid w \in \Sigma^*\}.$$

We solve the problem whether HCS_k includes non-regular or non-context-free languages which was asked in [17]. Furthermore, we will show that the result also holds if we consider the iterated one-sided hairpin completion.

Let us recall that, as we are treating the unbounded hairpin completion now, for the usual factorization $\gamma\alpha\beta\overline{\alpha}\overline{\gamma}$ of a hairpin completion, the length of the factor γ is not bounded by a constant anymore. By the results of the previous section it is obvious, that the possibility of creating arbitrary long prefixes and suffixes plays an essential role in following proof.

Theorem 5.1. *The iterated one- and two-sided hairpin completions of a singleton are in* NL *but not context-free, in general.*

Proof. The membership to NL follows by the fact that NL is closed under iterated bounded hairpin completion, which has been proved in [1]. For convenience, we give a sketch of the proof, here.

Consider a language $L \in$ NL. The iterated hairpin completion $\mathcal{H}_k^*(L)$ can be accepted by a non-deterministic Turing machine that works as follows. We use two pointers i and j which mark the beginning and the end of a factor of the input w, respectively. By $w(i, j)$ we denote the factor beginning at position i and ending at position j.

1. We start with $i = 1$ and $j = |w|$.

2. Non-deterministically either continue with step 3 or skip to step 5.

3. Either guess i' such that $i < i' < j$ and verify that $w(i, j)$ is a left hairpin completion of $w(i', j)$ or guess j' such that $i < j' < j$ and verify that $w(i, j)$ is a right hairpin completion of $w(i, j')$. If the verification is successful, continue with $i = i'$ (resp. $j = j'$).

4. Repeat step 2.

5. Accept if and only if $w(i, j) \in L$.

Obviously, this Turing machine accepts $\mathcal{H}_k^*(L)$. In order to perform step 1-4, we only have to store some pointers on the input word; this can be done in $\log|w|$ space. Since $L \in \text{NL}$ step 5 can be performed in $\log|w|$ space, too, and hence $\mathcal{H}_k^*(L) \in \text{NL}$.

For the one-sided hairpin completion $\mathcal{RH}_k(L)$ we can use almost the same algorithm. The only difference is that the pointer i always is 1.

Now, let $\Sigma = \{a, \bar{a}, b, \bar{b}, c, \bar{c}\}$, $\alpha = a^k$, and

$$w = \alpha b \alpha \bar{\alpha} \alpha c \bar{\alpha}.$$

We will prove that $\mathcal{H}_k^*(\{w\})$ and $\mathcal{RH}_k^*(\{w\})$ are not context-free.

Since context-free languages are closed under intersection with regular languages, it suffices to show for a regular language R that the intersections $R \cap \mathcal{H}_k^*(\{w\})$ and $R \cap \mathcal{RH}_k^*(\{w\})$ are not context-free. Let $u = \bar{b}\bar{\alpha}$ and $v = \alpha\bar{\alpha}\bar{b}\bar{\alpha}$. Note that $\bar{u}\alpha \leq \bar{v}\alpha \leq w$. Define

$$R = wu^+v\bar{u}^+\overline{w}\bar{u}^+\overline{w}$$

and consider a word $z \in R$:

$$z = \underbrace{\alpha b \alpha \bar{\alpha}\alpha c \bar{\alpha}}_{w}\, \underbrace{(\bar{b}\bar{\alpha})^r}_{u^r}\, \underbrace{\alpha\bar{\alpha}\bar{b}\bar{\alpha}}_{v}\, \underbrace{(\alpha b)^s}_{\bar{u}^s}\, \underbrace{\alpha\bar{c}\bar{\alpha}\alpha\bar{\alpha}\bar{b}\bar{\alpha}}_{\overline{w}}\, \underbrace{(\alpha b)^t}_{\bar{u}^t}\, \underbrace{\alpha\bar{c}\bar{\alpha}\alpha\bar{\alpha}\bar{b}\bar{\alpha}}_{\overline{w}}$$

with $r, s, t \geq 1$. At first, note that w is a prefix of z and it does not occur as another factor in z (there is only one c in z). Thus, if z belongs to $\mathcal{H}_k^*(\{w\})$, it must be an iterated right hairpin completion of w and hence

$$R \cap \mathcal{H}_k^*(\{w\}) = R \cap \mathcal{RH}_k^*(\{w\}).$$

Next, we will show that z is an iterated hairpin completion of w if and only if $r = s = t$. The proof is a straight forward construction of z. We try to find a sequence $w = w_0, w_1, \ldots, w_n = z$ for some $n \geq 0$ where $w_i \neq w_{i-1}$ is a right hairpin completion of w_{i-1} for $1 \leq i \leq n$. This implies that every w_i is a prefix of z.

Fortunately, for each of the words $w_0, \ldots, w_{r+1}$ there is exactly one choice which satisfies these conditions:

$$
\begin{aligned}
w_0 &= w &&= \alpha b \alpha \bar{\alpha}\alpha c \bar{\alpha}\\
w_1 &= wu &&= \alpha b \alpha \bar{\alpha}\alpha c \bar{\alpha}\bar{b}\bar{\alpha}\\
w_2 &= wu^2 &&= \alpha b \alpha \bar{\alpha}\alpha c \bar{\alpha}(\bar{b}\bar{\alpha})^2\\
&\ \ \vdots && \ \ \vdots\\
w_r &= wu^r &&= \alpha b \alpha \bar{\alpha}\alpha c \bar{\alpha}(\bar{b}\bar{\alpha})^r\\
w_{r+1} &= wu^r v &&= \alpha b \alpha \bar{\alpha}\alpha c \bar{\alpha}(\bar{b}\bar{\alpha})^r\alpha\bar{\alpha}\bar{b}\bar{\alpha}
\end{aligned}
$$

If $s \neq r$, none of the right hairpin completions of w_{r+1} is a prefix of z (except for w_{r+1} itself). Otherwise, we find exactly one right hairpin completion which satisfies the conditions:

$$w_{r+2} = wu^r v\bar{u}^r\overline{w} = \alpha b \alpha \bar{\alpha}\alpha c \bar{\alpha}(\bar{b}\bar{\alpha})^r\alpha\bar{\alpha}\bar{b}\bar{\alpha}(\alpha b)^r\alpha\bar{c}\bar{\alpha}\alpha\bar{\alpha}\bar{b}\bar{\alpha}.$$

The argument for the last step is the same. If and only if $t = r$, we find a prefix of z which is a right hairpin completion of w_{r+2} and this is $w_{r+3} = z$.

We conclude z is an iterated hairpin completion of w if and only if $r = s = t$ and hence

$$R \cap \mathcal{H}_k^*(\{w\}) = \{wu^r v\overline{u}^r \overline{w}\overline{u}^r\overline{w} \mid r \geq 1\}.$$

The intersection $R \cap \mathcal{H}_k^*(\{w\})$ belongs to a family of context-sensitive languages which are well known to be non-context-free. From this it follows that $\mathcal{H}_k^*(\{w\})$ and $\mathcal{RH}_k^*(\{w\})$ are non-context-free, too. $\square$

6 Final Remarks and Open Problems

We proved that language classes which have very basic closure properties are closed under iterated bounded hairpin completion. With the techniques used in our proof we obtain a better insight on the structure of the iterated bounded hairpin completion. This might help to design new algorithms which decide the membership of a word to the iterated bounded hairpin completion of a given language and also for the unbounded version since for a given word there is an implicit given length bound.

Another interesting problem regarding the hairpin completion is whether the iterated hairpin completion of two languages have a common element. Even for two given singletons it is not known, if this problem is decidable at all, see [17]. The result of Section 5 proves that this is a non-trivial question. However, in the bounded case we can decide this problem for two regular languages now. We just need to create the NFAs and test whether the intersection is empty. As the size of the NFAs is quite large with respect to the length bounds, this does not seem to be the best way to decide the problem.

We proved the existence of non-context-free languages in the language class HCS. Here, two new questions arise naturally:

1. Does a singleton exist whose iterated hairpin completion is context-free but not regular?

2. Can we decide for a given singleton whether its iterated hairpin completion is non-regular (or non-context-free)?

References

[1] D. Cheptea, C. Martín-Vide, and V. Mitrana. A new operation on words suggested by DNA biochemistry: Hairpin completion. *Transgressive Computing*, pages 216–228, 2006.

[2] R. Deaton, R. Murphy, M. Garzon, D. Franceschetti, and S. Stevens. Good encodings for DNA-based solutions to combinatorial problems. *Proc. of DNA-based computers DIMACS Series*, 44:247–258, 1998.

[3] V. Diekert, S. Kopecki, and V. Mitrana. On the hairpin completion of regular languages. In M. Leucker and C. Morgan, editors, *ICTAC*, volume 5684 of *Lecture Notes in Computer Science*, pages 170–184. Springer, 2009.

[4] M. Garzon, R. Deaton, P. Neathery, R. Murphy, D. Franceschetti, and E. Stevens. On the encoding problem for DNA computing. *The Third DIMACS Workshop on DNA-Based Computing*, pages 230–237, 1997.

[5] M. Garzon, R. Deaton, L. Nino, S. Stevens Jr., and M. Wittner. Genome encoding for DNA computing. *Proc. Third Genetic Programming Conference*, pages 684–690, 1998.

[6] M. Hagiya, M. Arita, D. Kiga, K. Sakamoto, and S. Yokoyama. Towards parallel evaluation and learning of boolean μ-formulas with molecules. In *Second Annual Genetic Programming Conf.*, pages 105–114, 1997.

[7] J. E. Hopcroft and J. D. Ulman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[8] M. Ito, P. Leupold, F. Manea, and V. Mitrana. Bounded hairpin completion. *Information and Computation*, In Press, Accepted Manuscript:–, 2010.

[9] M. Ito, P. Leupold, and V. Mitrana. Bounded hairpin completion. In *LATA '09: Proceedings of the 3rd International Conference on Language and Automata Theory and Applications*, pages 434–445, Berlin, Heidelberg, 2009. Springer-Verlag.

[10] L. Kari, S. Konstantinidis, E. Losseva, P. Sosík, and G. Thierrin. Hairpin structures in DNA words. In A. Carbone and N. A. Pierce, editors, *DNA*, volume 3892 of *Lecture Notes in Computer Science*, pages 158–170. Springer, 2005.

[11] L. Kari, K. Mahalingam, and G. Thierrin. The syntactic monoid of hairpin-free languages. *Acta Inf.*, 44(3-4):153–166, 2007.

[12] S. Kopecki. On the iterated hairpin completion. In Y. Gao, H. Lu, S. Seki, and S. Yu, editors, *Developments in Language Theory*, volume 6224 of *Lecture Notes in Computer Science*, pages 438–439. Springer Berlin / Heidelberg, 2010.

[13] P. M. Lewis, R. E. Stearns, and J. Hartmanis. Memory bounds for recognition of context-free and context-sensitive languages. In *Proceedings of the 6th Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1965)*, FOCS '65, pages 191–202, Washington, DC, USA, 1965. IEEE Computer Society.

[14] F. Manea, C. Martín-Vide, and V. Mitrana. On some algorithmic problems regarding the hairpin completion. *Discrete Applied Mathematics*, 157(9):2143–2152, 2009.

[15] F. Manea and V. Mitrana. Hairpin completion versus hairpin reduction. In S. B. Cooper, B. Löwe, and A. Sorbi, editors, *CiE*, volume 4497 of *Lecture Notes in Computer Science*, pages 532–541. Springer, 2007.

[16] F. Manea, V. Mitrana, and T. Yokomori. Two complementary operations inspired by the DNA hairpin formation: Completion and reduction. *Theor. Comput. Sci.*, 410(4-5):417–425, 2009.

[17] F. Manea, V. Mitrana, and T. Yokomori. Some remarks on the hairpin completion. *Int. J. Found. Comput. Sci.*, 21(5):859–872, 2010.

[18] K. Sakamoto, D. Kiga, K. Komiya, H. Gouzu, S. Yokoyama, S. Ikeda, and M. Hagiya. State transitions by molecules, 1998.

[19] E. Winfree. Whiplash PCR for O(1) computing. In *University of Pennsylvania*, pages 175–188, 1998.